AN OUTLINE OF THE *SZEHAT* MACHINE TRANSLATION SYSTEM

Károly Fábricz

0. Preliminary Remarks.

The SZEHAT Machine Translation Project was launched in the fall of 1985. At that time it was started as a small-scale project aimed at experimentation in the field of computational linguistics in general, and English-Hungarian machine translation, in particular.

Then the research group consisted of five linguists and three computer specialists, employed at different institutions of the JATE University of Szeged. Since, unfortunately, the project failed to receive the financial support from government sources, the research had to be reorganized both in its architecture and the staff taking part in the programme. Regardless of the desperate efforts to maintain the enthusiasm of researchers, some of our colleagues decided to turn to questions not related to machine translation.

Consequently, a new computational infrastructure had to be found and also other specialists had to be involved. The reorganisation of SZEHAT in early 1987 led to a substantial modification of both the computational paradigm and the system architecture.

Previous research was based on a PC-LISP environment with a view to the possible application of currently available software for NLP in general. Sample parsers were constructed and debugged, research into the construction of a machine dictionary was taking place, but,

more characteristically, questions of transfer and Hungarian output generation were investigated.

The reorganisation of SZEHAT in the beginning of 1987 has led to the formation of a research group comprising specialists from two different institutions:the Humanities Faculty of the JATE University was responsible for issues of the theory of translation, as well as of the description of English and Hungarian (the latter being in itself a formidable task, since, up to date for Hungarian there exist no complete formal descriptions), and, on the other hand, the Cybernetics Laboratory provided some help both in hardware and assistance.

The other institution to take part in SZEHAT was the Research Team for the Theory of Automata at the Szeged Affiliation of the Hungarian Academy of Sciences which took the responsibility of collaborating in software design.

1. An overview of SZEHAT.

At present, SZEHAT (Szeged University English-Hungarian Automatic Translation) can be viewed as a prototype system hopefully close to being able to run on our local IBM PC XT/AT network at the JATE Humanities Faculty. It will, however, remain an experimental system for a long time. It seems probable that the mainstream of future work will be oriented on the development of the lexicon and transfer rather than on modifying system architecture on the whole. The latter is presented in some detail below:

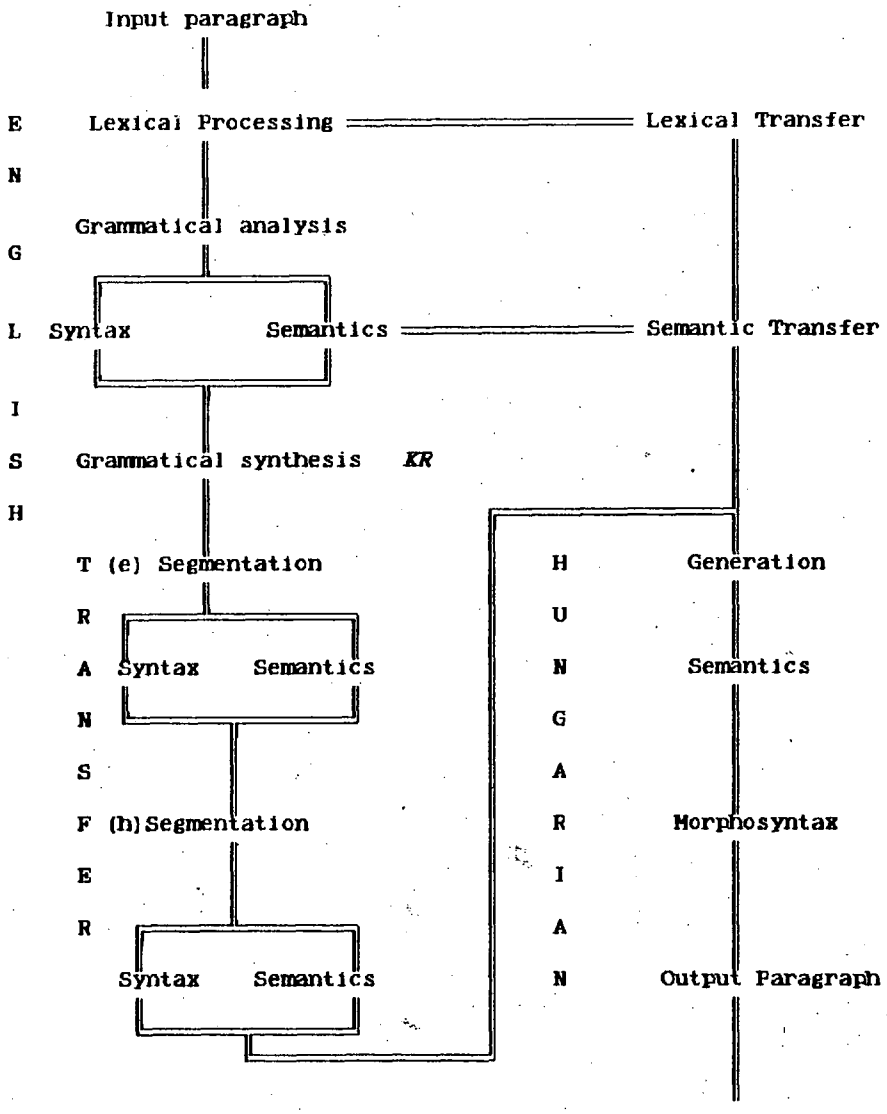

Table 1. The Architecture of SZEHAT

As illustrated in the table above, the process of translation is split up into a sequence of modules consisting of:

- reading the input paragraph
- lexical processing
- grammatical analysis
- grammatical synthesis (KR)
- segmentation I.
- segmentation II.
- semantic generation
- morphosyntactic generation
- writing the output paragraph.

2.0. General system description.

Observing the order of the modules listed above, the present state of research into the construction of SZEHAT may be briefly summarized as follows.

2.1. Chosing a paragraph to be the unit of translation is motivated by the inadequacy of treating such textual phenomena as pronominalization or reference on a purely sentence-by-sentence level. Some investigations established on sentence-based translation have shown that e.g. referential assignment, in the case

of English--Hungarian translation, is of crucial importance - not only in its own right but also due to a specific feature of the Hungarian language, whereby proper indexation affects not only the class of pronouns but phenomena on the morphological level as well (e.g. choice of verb-forms, conjugation types, or prefix + stem inversion).

A text-oriented approach also has other theoretical and practical advantages as well. We shall return to this question under 2.7. below.

2.2. Lexical processing at present is limited to a narrow range of not more than 1.000 English dictionary entries and about as many Hungarian lexemes. Nevertheless, this amount covers much of the basic word stock most commonly used in scientific papers. Selection of words for this basic word stock was carried out by way of a simple sorter programme (QSORT, as described by Wirth and implemented in Pascal) on a corpus of some 200.000 words typed in from different scientific articles with a thematic span ranging from papers on linguistics, chemistry, biology, computer technology, geography, and information technology.

It would seem appropriate to throw some light on the construction of the lexicon. SZEHAT is supposed to carry out machine translation on a wide range of scientific topics. This intention is reflected in the modular build-up of the lexicon:

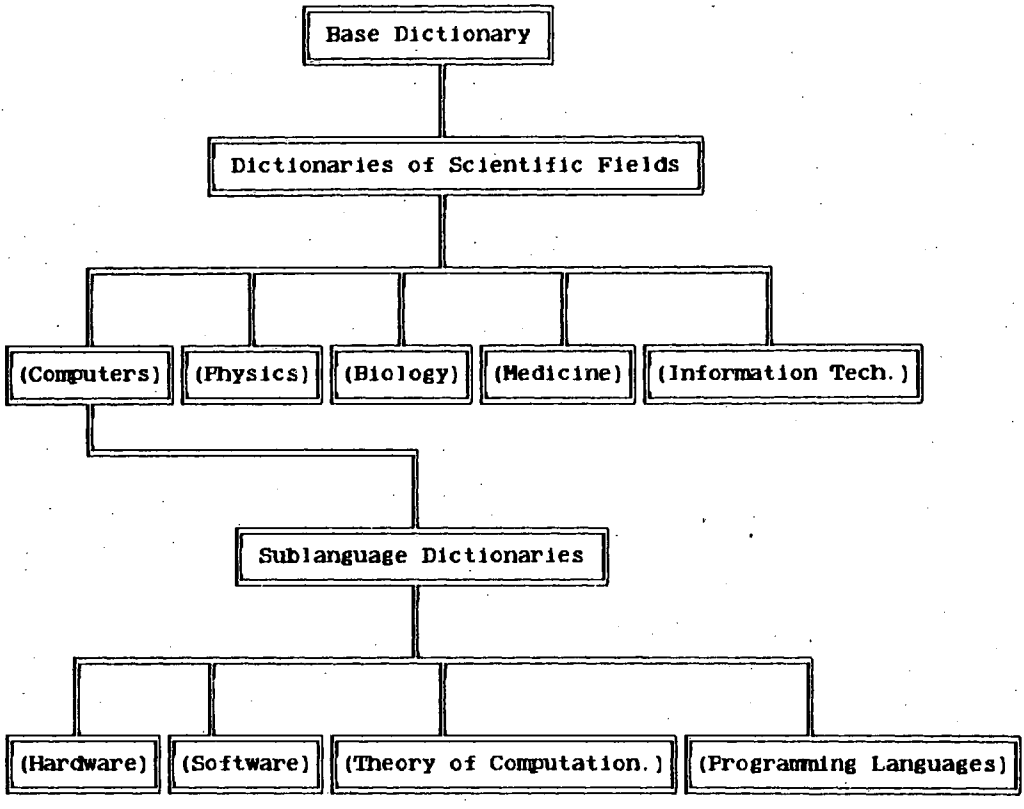

Table 2. The Construction of the Lexicon

When the input paragraph is typed in the entries go through dictionary look-up and are supplied with their syntactic and semantic codes for subsequent processing, along with a pointer to their potential Hungarian counterparts. When a word is not found in the dictionary of basic words, it is passed to a sublexicon of scientific fields (for instance, that of computation) from where it can proceed to one of the sublanguage dictionaries (e.g. programming languages).

The choice of specific modules for a given article is based on the system for bibliographic cataloguing. The sublexicons and sublanguage dictionaries have yet to be constructed with the help of specialists from different branches of science.

At present, there is but limited morphological analysis for the English input. Due to the small size of the word stock, morphological phenomena are dealt with during dictionary look-up: the word-forms of lexemes are integrated under one heading, labelled with the head word, while the actual morphological information is embodied in the syntactic and/or semantic code.

Thus, a lexical entry in SZEHAT would look something like this:

```
STACKS [headword: STACK] ----------------> 0014 [pointer to
                                                 Hung. eq.]

[N9-110A-N-] [syntactic code]
[NT-421-ina] [semantic code]
```

Figure 1. A sample lexical entry in SZEHAT base dictionary.

A phrasal dictionary for the base lexicon has also been constructed. It is quite simple and works basically as a list-matching procedure: the word marked in the base lexicon as a possible candidate for coexisting with one or more words making up an idiom is matched against a list of words following it so as to identify it as a member of an idiom. If that fails, the procedure is terminated. If the list-matching succeeds the list in itself is supplied with the codes for subsequent analysis.

The phrasal dictionary contains not more than 100 items, and it can only handle phrases making up just one sentence constituent (an NP or a VP).

2.3. English grammatical analysis includes the application of a set of different parsers implemented basically in Pascal. English analysis is based on a modification of the LSP grammar of New York University as described in the book by Naomi Sager. Among the grammars available we chose this one for two reasons: on the one hand, its BNF component seemed to fit our PROF-LP language processor. On the other hand, a modification of its restriction grammar so as to meet the requirements of our attribute grammar appeared to be a not too formidable task.

The PROF-LP language processor was originally developed by the Research Group on the Theory of Automata at the Academy of Sciences. It is a programme package for writing attribute grammars and is marketed by the Research Group. It is also the language processor software used by the other Hungarian researchers currently investigating Russian-Hungarian machine translation (the MTA SZTAKI project).

PROF-LP is a program generator system for IBM XT/AT compatible computers. On the basis of an attribute grammar description, the system generates a Pascal program which can be viewed as the processor of the given specification. PROF-LP can be used for generating both one-pass and multi-pass language processors. A further advantage of the system is that a Pascal 8000 version is also available for IBM 360/370 computers, i.e. it runs on the university mainframe.

Grammatical analysis in SZEHAT is basically carried out following the rules specified in Sager's monograph. The PROF-LP attribute grammar makes it possible to divide English input processing into three phases to be integrated later on during grammatical synthesis. Thus, syntactic (BNF) and semantic (restriction) analyses are built on each other, with lexical analysis terminating input processing. The latter results in an output providing a basis both for semantic transfer and grammatical synthesis.

2.4. Grammatical synthesis is a peculiar feature of SZEHAT aimed at providing opportunity for subsequent processing in a new dimension. In fact, this phase regains all results relevant for later operations.

As a result of previous analyses, synthesis makes it possible to rule out faulty solutions and to apply constituent-level,sentence-level and paragraph-level operations. These operations include different kinds of techniques for obtaining an appropriate Hungarian paragraph. Among them the most significant are the application of knowledge representation and various types of a procedure named segmentation. Syntactic synthesis serves as a basis for segmentation (see below), while semantic synthesis

results in a semantic structure that is carried over to the phase named "semantic transfer" together with the result of the technique boastfully called "knowledge representation".

2.5. Knowledge Representation in SZEHAT at the present state of affairs is restricted to reducing the contents of the sentences containing a paragraph to a simpler semantic representation. On the one hand, it implies establishing so-called "base relations" demonstrating different semantic operations such as inclusion, paraphrasing, description, illustration etc. On the other hand, the semantic operations identified are directly "translated" into Hungarian in order to obtain a frame for the output paragraph.

In other words, KR is at present used to generate Hungarian sentences with a semantic structure identical to that of the English input. The term "KR" is used here in a rather unusual sense. In fact, with its help we try to make use of the observation that a scientific article is basically constructed according to cliches whose representation is to some extent language-specific. The advantage of applying a quasi-KR basis for semantic transfer is manifest not only in acquiring some better translation but also in filtering out most of the features characteristic of English input, though not apparent in Hungarian.

Thus, KR is responsible for converting the majority of passive constructions into active ones in Hungarian and also for imposing restrictions on possible frame structures.

At present, this module handles only a small set of semantic features. Nevertheless, its application seems essential for the system to run well. The module consists of a pattern recognizer, a Hungarian pattern generator, and a frame generator. The Hunga-

rian patterns are stored in a synonymous arrangement, i.e. all semantic relations are represented by a set of Hungarian patterns from which the suitable one is selected according to the degree of lexical coincidence.

2.6 English and Hungarian Segmentations represent a syntactic-semantic transfer device that takes into account the configurational difference between English and Hungarian. During segmentation the English structure arrived at in "2.4." is ascribed a configuration and the latter is transformed into Hungarian -- a non-configurational (or semi-configurational) language.

The Segmentation Module breaks the input into constitutional units recognizing such phenomena as topicalization, wh-movement, co-indexation etc. in order to map the given configuration onto the Hungarian output. Syntactic segmentation is also responsible for handling such English phenomena as tenses of the verb (Hungarian has a much simpler system of tenses), multiple possessive constructions (which usually begin with a reverse order of elements) etc. Semantic segmentation is important, among others, for defining Hungarian verbal conjugational types on the basis of the evaluation of an English object. It means that specific features of Hungarian translation are accounted for before actual generation of the Hungarian output comes into force.

2.7. The final stage of the translational process will be the generation of the Hungarian output. It is going to be carried out on the basis of the following parts:

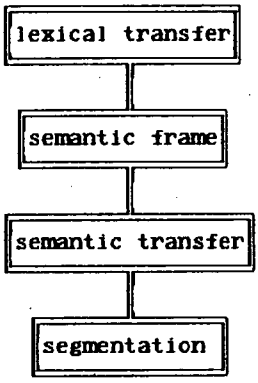

Table 3. Hungarian output generation

Semantic frame and semantic transfer represent the starting-point for Hungarian generation. Semantic frame reflects that of the English input, while semantic transfer provides clues for criteria of well-formedness on constituent level.

Lexical transfer takes place only when all the restrictions for Hungarian sentence construction have been imposed.

The last stage preceding morphosyntactic generation is the application of the results of segmentation. The aim of segmentation is to produce the constitutional organization of the Hungarian paragraph with respect to configurationality in Hungarian.

It is at this stage that the advantage of segmentation (should we call it 'word-order generation'?) becomes manifest.

We find that morphosyntactic generation is thus confined to a substantial morphological generation.

3. To sum up, SZEHAT is an experimental English-Hungarian MT system currently being developed with the joint effort of specialists (or budding specialists) from different institutions of Szeged. The characteristic features of SZEHAT are:

a) linguistic analysis is carried out with the help of an attribute grammar;
b) the unit of analysis is the paragraph;
c) SZEHAT is based on modular transfer;
d) Hungarian generation partly rests on a KR-like semantic representation;
e) SZEHAT is being implemented on a configuration of IBM XT/AT computers, the programming languages employed being Pascal and Prolog.